



```
DDDDDDDD  RRRRRRR  7777777  555555555  000000
DDDDDDDD  RRRRRRR  7777777  555555555  000000
DD      DD  RR      RR      77      55      00      00
DD      DD  RR      RR      77      55      00      00
DD      DD  RR      RR      77      555555  00      0000
DD      DD  RR      RR      77      555555  00      0000
DD      DD  RRRRRRR  77      55      00      00      00
DD      DD  RRRRRRR  77      55      00      00      00
DD      DD  RR  RR  77      55      0000      00
DD      DD  RR  RR  77      55      0000      00
DD      DD  RR  RR  77      55      00      00
DD      DD  RR  RR  77      55      00      00
DDDDDDDD  RR      RR  77      555555  000000
DDDDDDDD  RR      RR  77      555555  000000
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```



E 9  
16-Sep-1984 00:00:59  
5-Sep-1984 13:53:17

VAX-11 FORTRAN V3.4-56  
DISK\$VMSMASTER:[ERF.SRC]DR750.FOR;1

Page 1

```
0001 Subroutine DR750 (lun)
0002 C
0003 C Version: 'V04-000'
0004 C
0005 C*****
0006 C*
0007 C* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0008 C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0009 C* ALL RIGHTS RESERVED.
0010 C*
0011 C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0012 C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0013 C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0014 C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0015 C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0016 C* TRANSFERRED.
0017 C*
0018 C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0019 C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0020 C* CORPORATION.
0021 C*
0022 C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0023 C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0024 C*
0025 C*
0026 C*****
0027 C
0028 C
0029 C
0030 C Author: Sharon Reynolds Creation date: 21-July-1981
0031 C
0032 C Functional Description:
0033 C
0034 C This module decodes and outputs the error information for the
0035 C DR750.
0036 C
0037 C Modified by:
0038 C
0039 C V03-004 SAR0219 Sharon A. Reynolds, 28-Mar-1984
0040 C Changed the call to UCBSL_GWNUIC to ORBSL_OWNER.
0041 C
0042 C V03-003 SAR0111 Sharon A. Reynolds, 23-Jun-1983
0043 C Changed the carriage control in the 'format' statements
0044 C for use with ERF.
0045 C
0046 C V03-002 SAR0039 Sharon A. Reynolds, 8-Jun-1983
0047 C Removed brief/cryptic support.
0048 C
0049 C v03-001 BP0001 Brian Porter, 20-AUG-1982
0050 C Minor edit.
0051 C**
0052 C--
0053 C
0054 C Include 'SRC$:DR32COM.FOR /NOLIST'
0105 C Include 'SRC$:DEVERR.FOR /NOLIST'
0206 C Include 'SRC$:MSGHDR.FOR /NOLIST'
0265
```

```

0266      Byte      lun
0267
0268      Integer*4   dr_dcr
0269      Integer*4   dr_utl
0270      Integer*4   dr_wcs_adr
0271      Integer*4   dr_wcs_dat
0272      Integer*4   dr_cmi_adr
0273      Integer*4   dr_cmi_bcr
0274      Integer*4   dr_di_bcr
0275      Integer*4   dr_dsl
0276      Integer*4   dr_sbr
0277      Integer*4   dr_gbr
0278      Integer*4   dr_cb(1:3)
0279      Integer*4   dr_bb(1:3)
0280      Integer*4   dr_gpr(0:22)
0281
0282      Integer*4   silo_byte_cnt
0283      Integer*4   compressf
0284      Integer*4   compressc
0285      Integer*4   compress4
0286      Integer*4   field
0287      Integer*4   field1
0288
0289      Real        float
0290
0291      Parameter    dr_750 = 48
0292
0293      Character*24 dr_dcr1(11:12)
0294      Character*17 dr_dcr2(14:20)
0295      Character*32 dr_dcr3(22:31)
0296      Character*24 bus_func(0:7)
0297
0298      C
0299      C      Make the necessary information in the error log buffer available
0300      C      to this module.
0301      C
0302      Equivalence  (dr_dcr,EMBSL_DV_REGSAV(0))
0303      Equivalence  (dr_utl,EMBSL_DV_REGSAV(1))
0304      Equivalence  (dr_wcs_adr,EMBSL_DV_REGSAV(2))
0305      Equivalence  (dr_wcs_dat,EMBSL_DV_REGSAV(3))
0306      Equivalence  (dr_cmi_adr,EMBSL_DV_REGSAV(5))
0307      Equivalence  (dr_cmi_bcr,EMBSL_DV_REGSAV(6))
0308      Equivalence  (dr_di_bcr,EMBSL_DV_REGSAV(7))
0309      Equivalence  (dr_dsl,EMBSL_DV_REGSAV(8))
0310      Equivalence  (dr_sbr,EMBSL_DV_REGSAV(9))
0311      Equivalence  (dr_gbr,EMBSL_DV_REGSAV(10))
0312      Equivalence  (dr_cb,EMBSL_DV_REGSAV(11))
0313      Equivalence  (dr_bb,EMBSL_DV_REGSAV(14))
0314      Equivalence  (dr_gpr,EMBSL_DV_REGSAV(17))
0315
0316
0317      Data  dr_dcr1(11)  /*DATA INTERCONNECT STALL*/
0318      Data  dr_dcr1(12)  /*JAM*/
0319
0320
0321      Data  dr_dcr2(14)  /*NO CMI RESPONSE*/
0322      Data  dr_dcr2(15)  /*RDS ERROR*/

```



```

0323 Data dr_dcr2(16) /*CRD ERROR*/
0324 Data dr_dcr2(17) /*HALT*/
0325 Data dr_dcr2(18) /*DCR ABORT*/
0326 Data dr_dcr2(19) /*PACKET INTERRUPT*/
0327 Data dr_dcr2(20) /*INTERRUPT ENABLE*/
0328
0329 Data dr_dcr3(22) /*POWER UP*/
0330 Data dr_dcr3(23) /*POWER DOWN*/
0331 Data dr_dcr3(24) /*EXTERNAL ABORT*/
0332 Data dr_dcr3(25) /*FORCE CI PARITY ERROR*/
0333 Data dr_dcr3(26) /*FORCE DI PARITY ERROR*/
0334 Data dr_dcr3(27) /*ENABLE DI PARITY ERROR ABORT*/
0335 Data dr_dcr3(28) /*LOCAL PARITY ERROR*/
0336 Data dr_dcr3(29) /*CI PARITY ERROR*/
0337 Data dr_dcr3(30) /*DI PARITY ERROR*/
0338 Data dr_dcr3(31) /*PARITY ERROR*/
0339
0340 Data bus_func(0) /*READ*/
0341 Data bus_func(1) /*READ LOCK*/
0342 Data bus_func(2) /*ILLEGAL FUNCTION*/
0343 Data bus_func(3) /*ILLEGAL FUNCTION*/
0344 Data bus_func(4) /*WRITE*/
0345 Data bus_func(5) /*WRITE UNLOCK*/
0346 Data bus_func(6) /*WRITE VECTOR*/
0347 Data bus_func(7) /*ILLEGAL FUNCTION*/
0348
0349
0350 C
0351 C Force a top of form to occur and construct the report header.
0352 C
0353 Call FRCTOF (lun)
0354
0355 Call DHEAD1 (lun,'DR32')
0356
0357 C
0358 C Decode and output the DR750 Control Register.
0359 C
0360 Call LINCHK (lun,2)
0361 Write (lun,20) dr_dcr
0362 20 Format (/,' ',T8,'DR DCR',T24,Z8.8)
0363
0364 Field = LIB$EXTZV (0,8,dr_dcr)
0365
0366 Call LINCHK (lun,1)
0367 If (field .ne. dr_750) then
0368
0369 Write (lun,25)
0370 25 Format (' ',T40,'ADAPTER NOT DR750')
0371 Else
0372
0373 Write (lun,30)
0374 30 Format (' ',T40,'ADAPTER IS DR750')
0375 Endif
0376
0377 Call OUTPUT (lun,dr_dcr,dr_dcr1,11,11,12,'0')
0378
0379 Call OUTPUT (lun,dr_dcr,dr_dcr2,14,14,20,'0')

```

```
0380
0381      Call OUTPUT (lun,dr_dcr,dr_dcr3,22,22,31,'0')
0382
0383      C
0384      C      Decode and output the DR750 Utility Register.
0385      C
0386      Call LINCHK (lun,1)
0387      Write (lun,40) dr_utl
0388      40      Format (' ',T8,'DR UTL',T24,Z8.8)
0389
0390      Field = LIB$EXTZV (0,8,dr_utl)
0391
0392      C
0393      C      See DR780 listing for explanation of speed calculation.
0394      C
0395
0396      If (field .le. 'FB'X) then
0397
0398      Field = JIDIM (256,field)
0399
0400      Float = FLOATJ(field)
0401
0402      float = 40/float
0403
0404      float = (1024*1024) / ((float*1024*1024)/4)
0405
0406      Call LINCHK (lun,2)
0407      Write (lun,45) float
0408      45      Format (' ',T40,'PERIOD = '
0409      1 F<COMPRESSF (float,1)>.1,' MICRO/SEC')
0410
0411      Float = FLOATJ(field)
0412
0413      float = (40/float)*1024
0414
0415      Write (lun,50) float
0416      50      Format (' ',T40,'TRANSFER = '
0417      1 F<COMPRESSF (float,2)>.2,' KB/SEC')
0418
0419      Endif
0420
0421      Field = LIB$EXTZV (11,1,dr_utl)
0422
0423      Call LINCHK (lun,1)
0424      If (field .eq. 1) then
0425
0426      Write (lun,60)
0427      60      Format (' ',T40,'MICRO-CODE LOADED AND VALID')
0428      Endif
0429
0430      Field = LIB$EXTZV (8,3,dr_utl)
0431      Field1 = LIB$EXTZV (12,4,dr_utl)
0432
0433      Call LIB$INSV (field,0,3,silo_byte_cnt)
0434      Call LIB$INSV (field1,3,4,silo_byte_cnt)
0435
0436      Call LINCHK (lun,2)
```



```

0437 Write (lun,70) silo_byte_cnt
0438 70 Format ('',T40,'SICO BYTE COUNT = ',
0439 1 i<compress4 (silo_byte_cnt)>,'.')
0440
0441 C
0442 C
0443 C Output the DR750 WCS Registers.
0444 C
0445 Call LINCHK (lun,2)
0446 Write (lun,80) dr_wcs_adr
0447 80 Format ('',T8,'DR WCS',T24,Z8.8,/,
0448 1 T40,'WCS ADDRESS REGISTER')
0449
0450 Call LINCHK (lun,2)
0451 Write (lun,85) dr_wcs_dat
0452 85 Format ('',T8,'DR WCS',T24,Z8.8,/,
0453 1 T40,'WCS DATA REGISTER')
0454
0455 C
0456 C
0457 C Decode and output the DR750 CMI Address Register.
0458 C
0459 Call LINCHK (lun,1)
0459 Write (lun,100) dr_cmi_adr
0460 100 Format ('',T8,'DR-BA',T24,Z8.8)
0461
0462 C
0463 C
0464 C This is the CMI address
0465 C
0466 Field = LIB$EXTZV (0,24,dr_cmi_adr)
0467
0468 C
0469 C
0470 C Clear bits 0:1 of the CMI address
0471 C
0472 Call LIB$INSV (0,0,2,field)
0473
0474 C
0475 C
0476 C Call LINCHK (lun,1)
0477 Write (lun,105) field
0478 105 Format ('',T40,'LONGWORD ADDRESS (HEX) = ',
0479 1 Z6)
0480
0481 C
0482 C
0483 C This is the CMI bus function.
0484 C
0485 Field = LIB$EXTZV (25,3,dr_cmi_adr)
0486
0487 C
0488 C
0489 C Call LINCHK (lun,1)
0490 Write (lun,110) bus_func(field)
0491 110 Format ('',T40,'CMI BUS FUNCTION = ',
0492 1 A<COMPRESSC (bus_func(field))> )
0493
0494 C
0495 C
0496 C This is the Byte mask (the dr750 will always read 32 bits (longwords)
0497 C but can mask which bytes to write).
0498 C
0499 Field = LIB$EXTZV (28,4,dr_cmi_adr)
0500

```

```
0494
0495 C
0496 C Decode and output the DR750 CMI Byte Count Register and the Data
0497 C Interconnect Byte Count Register.
0498 C
0499 Call LINCHK (lun,2)
0500 Write (lun,150) dr_cmi_bcr
0501 150 Format (' ',T8,'DR-BC',T24,Z8.8,/,
0502 1 T40,'CMI BYTE COUNT')
0503
0504 Call LINCHK (lun,2)
0505 Write (lun,160) dr_di_bcr
0506 160 Format (' ',T8,'DR-BC',T24,Z8.8,/,
0507 1 T40,'DATA INTERCONNECT BYTE COUNT')
0508
0509 C
0510 C Decode and output the DR750 Status Longword
0511 C
0512 Call LINCHK (lun,1)
0513 Write (lun,170) dr_dsl
0514 170 Format (' ',T8,'DR-DSL',T24,Z8.8)
0515
0516 Call OUTPUT (lun,dr_dsl,V1dr_sl,0,0,14,'0')
0517
0518 Field = LIB$EXTZV (16,8,dr_dsl)
0519
0520 If (field .ne. 0) then
0521
0522 Call LINCHK (lun,3)
0523 Write (lun,175) ('*', I = 0,32)
0524 175 Format (' ',T40,33A1,/,T47,'FAR-END STATUS BYTE',/)
0525
0526 Do 185, I = 0,4
0527
0528 Field1 = LIB$EXTZV (I,1,field)
0529
0530 If (field1 .ne. 0) then
0531
0532 Call LINCHK (lun,1)
0533 Write (lun,180) I
0534 180 Format (' ',T40,'FAR-END DEVICE STATUS BIT ',I1,'. SET')
0535 Endif
0536
0537 185 Continue
0538
0539 Call OUTPUT (lun,dr_dsl,V2dr_sl,21,21,23,'0')
0540
0541 Call LINCHK (lun,1)
0542 Write (lun,190) ('*', I = 0,32)
0543 190 Format (' ',T40,33A1)
0544 Endif
0545
0546 C
0547 C Output the DR750 System Base Register
0548 C
0549 Call LINCHK (lun,2)
0550 Write (lun,200) dr_sbr
```



```
0551 200 Format (' ',T8,'DR SBR',T24,Z8.8,/  
0552 1 T40,'DR SYSTEM BASE REG (PHY ADDR)')  
0553  
0554 C  
0555 C Output the DR750 Global Base Register  
0556 C  
0557 Call LINCHK (lun,2)  
0558 Write (lun,210) dr_gbr  
0559 210 Format (' ',T8,'DR-GBR',T24,Z8.8,/  
0560 1 T40,'DR GLOBAL BASE REG (VIRT ADDR)')  
0561  
0562 C  
0563 C Output the DR750 Command Block Information  
0564 C  
0565 Call LINCHK (lun,3)  
0566 Write (lun,220)  
0567 220 Format (' ',T8,'DR CURRENT COMMAND BLOCK',/)  
0568  
0569 Do 240 I = 1,3  
0570  
0571 Call LINCHK (lun,2)  
0572 Write (lun,230) I,dr_cb(I),vldr_cb(I)  
0573 230 Format (' ',T8,'DR CB',I1,T24,Z8.8,/  
0574 1 T40,A<COMPRESSC (vldr_cb(I))>)  
0575  
0576 240 Continue  
0577  
0578 C  
0579 C Output the DR750 Buffer Block Information  
0580 C  
0581 Call LINCHK (lun,3)  
0582 Write (lun,250)  
0583 250 Format (' ',T8,'DR CURRENT BUFFER BLOCK',/)  
0584  
0585 Do 270 I = 1,3  
0586  
0587 Call LINCHK (lun,2)  
0588 Write (lun,260) I,dr_bb(I),vldr_bb(I)  
0589 260 Format (' ',T8,'DR BB',I1,T24,Z8.8,/  
0590 1 T40,A<COMPRESSC (vldr_bb(I))>)  
0591  
0592 270 Continue  
0593  
0594 C  
0595 C Output the remaining portion of the WCS contents  
0596 C  
0597  
0598 Call LINCHK (lun,3)  
0599 Write (lun,280)  
0600 280 Format (' ',T8,'DR GENERAL PURPOSE REGISTERS',/)  
0601  
0602 Call LINCHK (lun,2)  
0603 Write (lun,290) dr_gpr(0)  
0604 290 Format (' ',T8,'DR-RO',T24,Z8.8,/  
0605 1 T40,'VIRTUAL ADDRESS')  
0606  
0607 Call LINCHK (lun,2)
```

```
0608      Write (lun,295) dr_gpr(1)
0609 295    Format (' ',T8,'DR R1',T24,Z8.8,/,
0610          1 T40,'PHYSICAL ADDRESS')
0611
0612      Field = LIB$EXTZV (0,16,dr_gpr(2))
0613
0614      Call LINCHK (lun,2)
0615      Write (lun,300) field
0616 300    Format (' ',T8,'DR R2',T24,Z4.4,/,
0617          1 T40,'DEVICE STATUS REGISTER')
0618
0619      Field = LIB$EXTZV (16,16,dr_gpr(2))
0620
0621      Call LINCHK (lun,2)
0622      Write (lun,305) field
0623 305    Format (T24,Z4.4,/,T40,'AREG')
0624
0625      Call LINCHK (lun,2)
0626      Write (lun,310) dr_gpr(3)
0627 310    Format (' ',T8,'DR R3',T24,Z8.8,/,
0628          1 T40,'CONTROL INTERCONNECT STATUS')
0629
0630      Call LINCHK (lun,2)
0631      Write (lun,315) dr_gpr(4)
0632 315    Format (' ',T8,'DR R4',T24,Z8.8,/,
0633          1 T40,'BUFFER ADDRESS')
0634
0635      Call LINCHK (lun,2)
0636      Write (lun,320) dr_gpr(5)
0637 320    Format (' ',T8,'DR R5',T24,Z8.8,/,
0638          1 T40,'RETURN ADDRESS, TRUE TEST RESULT')
0639
0640      Call LINCHK (lun,6)
0641      Do 330, I = 6,8
0642
0643      Write (lun,325) I,dr_gpr(I)
0644 325    Format (' ',T8,'DR R',I,T24,Z8.8,/,
0645          1 T40,'GENERAL PURPOSE REGISTER')
0646
0647 330    Continue
0648
0649      Field = LIB$EXTZV (0,16,dr_gpr(9))
0650
0651      Call LINCHK (lun,2)
0652      Write (lun,335) field
0653 335    Format (' ',T8,'DR R9',T24,Z4.4,/,
0654          1 T40,'ADDRESS OF CURRENT BUFFER')
0655
0656      Field = LIB$EXTZV (16,16,dr_gpr(9))
0657
0658      Call LINCHK (lun,2)
0659      Write (lun,340) field
0660 340    Format (T24,Z4.4,/,T40,'CURRENT BUFFER STATE')
0661
0662      Field = LIB$EXTZV (0,16,dr_gpr(10))
0663
0664      Call LINCHK (lun,2)
```



```
0665      Write (lun,345) field
0666 345      Format ('I,T8,'DR R10',T24,Z4.4,/,
0667          1 T40,'ADDRESS OF NEXT BUFFER')
0668
0669      Field = LIB$EXTZV (16,16,dr_gpr(10))
0670
0671      Call LINCHK (lun,2)
0672      Write (lun,350) field
0673 350      Format (T24,Z4.4,/,T40,'NEXT BUFFER STATUS')
0674
0675      Field = LIB$EXTZV (0,16,dr_gpr(11))
0676
0677      Call LINCHK (lun,2)
0678      Write (lun,355) field
0679 355      Format ('I,T8,'DR R11',T24,Z4.4,/,
0680          1 T40,'ADDRESS OF FREE BUFFER')
0681
0682      Field = LIB$EXTZV(16,16,dr_gpr(11))
0683
0684      Call LINCHK (lun,2)
0685      Write (lun,360) field
0686 360      Format (T24,Z4.4,/,T40,'FREE BUFFER STATUS')
0687
0688      Call LINCHK (lun,2)
0689
0690      Write (lun,365) dr_gpr(12)
0691 365      Format ('I,T8,'DR R12',T24,Z8.8,/,
0692          1 T40,'REMAINDER OF DI BYTE COUNT')
0693
0694      Call LINCHK (lun,2)
0695      Write (lun,370) dr_gpr(13)
0696 370      Format ('I,T8,'DR R13',T24,Z8.8,/,
0697          1 T40,'CONTROL INTERCONNECT BYTE COUNT')
0698
0699      Call LINCHK (lun,2)
0700      Write (lun,375) dr_gpr(14)
0701 375      Format ('I,T8,'DR R14',T24,Z8.8,/,
0702          1 T40,'CONTROL INTERCONNECT VA')
0703
0704      Call LINCHK (lun,2)
0705      Write (lun,380) dr_gpr(15)
0706 380      Format ('I,T8,'DR R15',T24,Z8.8,/,
0707          1 T40,'CONTROL INTERCONNECT PA')
0708
0709      Call LINCHK (lun,2)
0710      Write (lun,385) dr_gpr(16)
0711 385      Format ('I,T8,'DR R16',T24,Z8.8,/,
0712          1 T40,'SILO MASK')
0713
0714      Call LINCHK (lun,2)
0715      Write (lun,390) dr_gpr(17)
0716 390      Format ('I,T8,'DR R17',T24,Z8.8,/,
0717          1 T40,'CURRENTLY EXECUTING SILO COMMAND')
0718
0719      Call LINCHK (lun,2)
0720      Write (lun,400) dr_gpr(18)
0721 400      Format ('I,T8,'DR R18',T24,Z8.8,/,
```

```
0722      1 T40,'LAST VALUE LOADED, DI BYTE COUNT')
0723
0724      Call LINCHK (lun,2)
0725      Write (lun,405) dr_gpr(19)
0726 405    Format ('I,T8,'DR-R19',T24,Z8.8,/,
0727          1 T40,'LAST VALUE LOADED, CMI BYTE COUNT')
0728
0729      Call LINCHK (lun,2)
0730      Write (lun,410) dr_gpr(20)
0731 410    Format ('I,T8,'DR-R20',T24,Z8.8,/,
0732          1 T40,'CONTROL INTERCONNECT DATA WORD')
0733
0734      Call LINCHK (lun,2)
0735      Write (lun,415) dr_gpr(21)
0736 415    Format ('I,T8,'DR-R21',T24,Z8.8,/,
0737          1 T40,'CONTROL INTERCONNECT DATA')
0738
0739      Call LINCHK (lun,2)
0740      Write (lun,420) dr_gpr(22)
0741 420    Format ('I,T8,'DR-R22',T24,Z8.8,/,
0742          1 T40,'CONTROL INTERCONNECT INFORMATION')
0743
0744      call linchk (lun,1)
0745
0746      write(lun,425)
0747 425    format('I, :)
0748
0749      call orb$l_owner (lun,emb$l_dv_ownuic)
0750
0751      call ucb$l_char (lun,emb$l_dv_char)
0752
0753      call ucb$w_sts (lun,emb$w_dv_sts)
0754
0755      call ucb$l_opcnt (lun,emb$l_dv_opcnt)
0756
0757      call ucb$w_errcnt (lun,emb$w_dv_errcnt)
0758
0759      if (emb$w_hd_entry .ne. 98) then
0760
0761      call linchk (lun,1)
0762
0763      write(lun,425)
0764
0765      call xfdriver_qio (lun,emb$w_dv_func)
0766
0767      call irp$w_bcmt (lun,emb$w_dv_bcmt)
0768
0769      call irp$w_boff (lun,emb$w_dv_boff)
0770
0771      call irp$l_pid (lun,emb$l_dv_rqpid)
0772
0773      call irp$q_iosb (lun,emb$l_dv_iosb1)
0774      endif
0775
0776      Return
0777      End
```



## PROGRAM SECTIONS

| Name                  | Bytes | Attributes                              |
|-----------------------|-------|-----------------------------------------|
| 0 \$CODE              | 3498  | PIC CON REL LCL SHR EXE RD NOWRT LONG   |
| 1 \$PDATA             | 2066  | PIC CON REL LCL SHR NOEXE RD NOWRT LONG |
| 2 \$LOCAL             | 1668  | PIC CON REL LCL NOSHR NOEXE RD WRT LONG |
| 3 DR32                | 600   | PIC OVR REL GBL SHR NOEXE RD WRT LONG   |
| 4 EMB                 | 512   | PIC OVR REL GBL SHR NOEXE RD WRT LONG   |
| Total Space Allocated | 8344  |                                         |

## ENTRY POINTS

| Address    | Type | Name  |
|------------|------|-------|
| 0-00000000 |      | DR750 |

## VARIABLES

| Address      | Type | Name            | Address    | Type | Name            |
|--------------|------|-----------------|------------|------|-----------------|
| 4-00000066   | I*4  | DR_CMI_ADR      | 4-0000006A | I*4  | DR_CMI_BCR      |
| 4-00000052   | I*4  | DR_DCR          | 4-0000006E | I*4  | DR_DI_BCR       |
| 4-00000072   | I*4  | DR_DSL          | 4-0000007A | I*4  | DR_GBR          |
| 4-00000076   | I*4  | DR_SBR          | 4-00000056 | I*4  | DR_UTL          |
| 4-0000005A   | I*4  | DR_WCS_ADR      | 4-0000005E | I*4  | DR_WCS_DAT      |
| 4-0000001C   | L*1  | EMBSB_DV_CLASS  | 4-00000010 | L*1  | EMBSB_DV_ERTCNT |
| 4-00000011   | L*1  | EMBSB_DV_ERTMAX | 4-0000003E | L*1  | EMBSB_DV_NAMLNG |
| 4-0000003A   | L*1  | EMBSB_DV_SLAVE  | 4-0000001D | L*1  | EMBSB_DV_TYPE   |
| 4-00000036   | I*4  | EMBSL_DV_CHAR   | 4-00000012 | I*4  | EMBSL_DV_IOSB1  |
| 4-00000016   | I*4  | EMBSL_DV_IOSB2  | 4-00000026 | I*4  | EMBSL_DV_MEDIA  |
| 4-0000004E   | I*4  | EMBSL_DV_NUMREG | 4-0000002E | I*4  | EMBSL_DV_OPCNT  |
| 4-00000032   | I*4  | EMBSL_DV_OWNUIC | 4-0000001E | I*4  | EMBSL_DV_RQPID  |
| 4-00000000   | I*4  | EMBSL_HD_SID    | 4-0000003F | CHAR | EMBST_DV_NAME   |
| 4-00000024   | I*2  | EMBSW_DV_BCNT   | 4-00000022 | I*2  | EMBSW_DV_BOFF   |
| 4-0000002C   | I*2  | EMBSW_DV_ERRCNT | 4-0000003C | I*2  | EMBSW_DV_FUNC   |
| 4-0000001A   | I*2  | EMBSW_DV_STS    | 4-0000002A | I*2  | EMBSW_DV_UNIT   |
| 4-00000004   | I*2  | EMBSW_HD_ENTRY  | 4-0000000E | I*2  | EMBSW_HD_ERRSEQ |
| 2-000002AC   | I*4  | FIELD           | 2-000002B0 | I*4  | FIELDT          |
| 2-000002B4   | R*4  | FLOAT           | 2-000002B8 | I*4  | I               |
| AP-00000004a | L*1  | LUN             | 2-000002A8 | I*4  | SILO_BYTE_CNT   |

## ARRAYS

| Address    | Type | Name     | Bytes | Dimensions |
|------------|------|----------|-------|------------|
| 2-000001E7 | CHAR | BUS_FUNC | 192   | (0:7)      |
| 4-0000008A | I*4  | DR_BB    | 12    | (3)        |
| 4-0000007E | I*4  | DR_CB    | 12    | (3)        |
| 2-00000000 | CHAR | DR_DCR1  | 48    | (11:12)    |
| 2-00000030 | CHAR | DR_DCR2  | 119   | (14:20)    |
| 2-000000A7 | CHAR | DR_DCR3  | 320   | (22:31)    |
| 4-00000096 | I*4  | DR_GPR   | 92    | (0:22)     |

|     |         |
|-----|---------|
| 512 | (0:511) |
| 420 | (0:104) |
| 8   | (2)     |
| 93  | (3)     |
| 96  | (3)     |
| 345 | (0:14)  |
| 66  | (21:23) |

| Address    | Label | Address    | Label | Address    | Label | Address    | Label | Address    | Label | Address    | Label |
|------------|-------|------------|-------|------------|-------|------------|-------|------------|-------|------------|-------|
| 1-00000054 | 20'   | 1-00000068 | 25'   | 1-00000081 | 30'   | 1-00000099 | 40'   | 1-000000AC | 45'   | 1-000000D1 | 50'   |
| 1-000000F5 | 60'   | 1-00000118 | 70'   | 1-0000013B | 80'   | 1-00000167 | 85'   | 1-00000190 | 100'  | 1-000001A2 | 105'  |
| 1-000001C5 | 110'  | 1-000001E6 | 150'  | 1-0000020B | 160'  | 1-0000023E | 170'  | 1-00000251 | 175'  | 1-00000274 | 180'  |
| **         | 185   | 1-0000029F | 190'  | 1-000002A9 | 200'  | 1-000002DE | 210'  | 1-00000314 | 220'  | 1-00000334 | 230'  |
| **         | 240   | 1-00000351 | 250'  | 1-00000370 | 260'  | **         | 270   | 1-0000038D | 280'  | 1-000003B1 | 290'  |
| 1-000003D7 | 295'  | 1-000003FE | 300'  | 1-0000042B | 305'  | 1-0000043A | 310'  | 1-0000046C | 315'  | 1-00000491 | 320'  |
| 1-000004C8 | 325'  | **         | 330   | 1-000004F8 | 335'  | 1-00000528 | 340'  | 1-00000547 | 345'  | 1-00000575 | 350'  |
| 1-00000592 | 355'  | 1-000005C0 | 360'  | 1-000005DD | 365'  | 1-0000060F | 370'  | 1-00000646 | 375'  | 1-00000675 | 380'  |
| 1-000006A4 | 385'  | 1-000006C5 | 390'  | 1-000006FD | 400'  | 1-00000735 | 405'  | 1-0000076E | 410'  | 1-000007A4 | 415'  |
| 1-000007D5 | 420'  | 1-0000080D | 425'  |            |       |            |       |            |       |            |       |

| Type | Name          | Type | Name       | Type | Name         | Type | Name          | Type | Name           |
|------|---------------|------|------------|------|--------------|------|---------------|------|----------------|
| I*4  | COMPRESS4     | I*4  | COMPRESSC  | I*4  | COMPRESSF    |      | DHEAD1        |      | FRCTOF         |
|      | IRPSQ_IOSB    |      | IRPSW_BCNT |      | IRPSW_BOFF   | I*4  | LIB\$EXTZV    |      | LIB\$INSV      |
|      | ORB\$\$_OWNER |      | OUTPUT     |      | UCB\$\$_CHAR |      | UCB\$\$_OPCNT |      | UCB\$\$_ERRCNT |
|      | XFDRIIVER_QIO |      |            |      |              |      |               |      | IRP\$\$_PID    |
|      |               |      |            |      |              |      |               |      | LINCHR         |
|      |               |      |            |      |              |      |               |      | UCB\$\$_STS    |



0001  
0002

## COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:DR750/OBJ=OBJ\$:DR750 MSRC\$:DR750

```

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)

```

```
/DEBUG=(NOSYMBOLS,TRACEBACK)
```

/STANDARD=(NOSYNTAX,NOSOURCE,FORM)

```
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
```

```

/FF77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

```

## COMPILATION STATISTICS

```
Run Time:          9.31 seconds
Elapsed Time:      20.52 seconds
Page Faults:       262
Dynamic Memory:    251 pages
```



0147

AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY